UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/699,144	10/31/2003	Dhruva Ranjan Chakrabarti	200313003-1	3438

22879 7590 06/20/2008

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
WU, JUNCHUN	
ART UNIT	PAPER NUMBER
2191	

NOTIFICATION DATE	DELIVERY MODE
06/20/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com
ipa.mail@hp.com

Office Action Summary	Application No.	Applicant(s)	
	10/699,144	CHAKRABARTI ET AL.	
	Examiner	Art Unit	
	JUNCHUN WU	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 March 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-8 and 10-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-8, and 10-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to the amendment filed on Mar. 14, 2008.

2. Claims 1, 8, 15, 16, 17 are amended.

3. Claims 1, 3-8, 10-15 are pending in this application.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1, 4, 5, 7, 8, 11, 12, and 14-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over by Ayers et al. ("Aggressive Inlining", 1997, ACM, hereafter "Ayers") and in view of Schmidt (US Patent No. 6,195,793 B1).

6. Per claim 1 (Currently amended)

Ayers discloses

- A method of compiling a computer program, the method composing:
- receiving a plurality of modules of source code (Fig.1).
- generating intermediate representations corresponding to the modules (Sec.2.1 1st Para. Lines 1-6).
- extracting a set of data from the intermediate representations to create an inliner summary for each module (sec.2.2 1st Para. Lines 1-4 & 3rd Para. Lines 1-5).

- after a call site is determined to be inlined: updating a call graph of the routines and call sites, and updating the inliner summaries throughout the call graph (Sec.2.3 The Last Para.).

But Ayers dose not discloses

- using the inliner summaries and a globally-sorted working-list based order in an inline analysis phase, without using the intermediate representations in the inline analysis phase, to determine which call sites in the modules are to be inlined by substituting code from a called module, wherein said globally-sorted working-list based order is dynamically updated during the inline analysis phase.
- Determining the call sites to be inlined involves proceeding only once through the call sites in said dynamically-updated globally-sorted working-list based order.

However, Schmidt implicitly discloses

- using the inliner summaries and a globally-sorted working-list based order in an inline analysis phase, without using the intermediate representations in the inline analysis phase, to determine which call sites in the modules are to be inlined by substituting code from a called module (col.3 lines 42~44 & lines 53-62).
- Determining the call sites to be inlined involves proceeding only once through the call sites (col.3 lines 42-53 *"A more global view of inlining effects is provided in order to select good inlining candidates while accurately spending the code bloat budget. First the best call sites at which to inline are estimated, based upon the execution frequencies of the call sites and the sizes of the called procedures. Then the call graph is processed starting from the leaves and working up. Each time an arc that was selected for inlining*

is encountered, the original bloat estimate is compared with the current size of the procedure, for example, incorporating sizes of any procedures that were inlined into the procedure. If the called procedure has been bloated beyond an acceptable limitation, it may be rejected for inlining.").

- dynamically updated the working list of call sites during the incline analysis phase (col.7 lines 31-38 *"If the call site exceeds the allowable growth at decision block 408, its priority is recalculated based on the bloat now known that will be incurred as indicated at a block 410. A best call site j residing in the AuxQueue is obtained as indicated at a block 412. Then checking to see whether the best call site j residing in the AuxQueue obtained at block 412 has a priority at least as great as the one being considering is performed as indicated at a decision block 414. "*).
- Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ayer's teachings by adding using the inliner summaries and a globally-sorted working-list based order in an inline analysis phase, without using the intermediate representations in the inline analysis phase, to determine which call sites in the modules are to be inlined by substituting code from a called module, wherein said globally-sorted working-list based order is dynamically updated during the inline analysis phase as taught by Schmidt in order to select good inlining candidates by an adaptive approach in accordance with features of preferred embodiment (col.3 lines 40-42).

7. Per claims 4 and 11 (Previously presented)

the rejection of claim 3 and 10 are incorporated respectively and Ayers discloses

- updating the inliner summaries comprises determining nodes and edges of the call graph that are affected by the inlining of the call site and updating those inliner summaries corresponding to the affected nodes and edges (Sec.2.3 The Last Para. Lines 5-8).

8. Per claims 5 and 12 (Previously presented)

the rejection of claim 4 and 11 are incorporated respectively and Ayers discloses

- the edge summaries include at least a call site execution count and a signature type (Sec.2.3 1st Para.).

9. Per claims 7 (Original)

the rejection of claim 1 is incorporated and Ayers further discloses

- the inline analysis phase is separate and distinct from an inline transformation phase (Sec. 2.3 1st & 2nd Para. (i.e. analysis phase for determining which call site is clonable); Sec.2.3 6th & 7th Para. (i.e. transformation phase for using the results of analysis phase to create clones and fix call sites).

10. Per claim 8 (Currently amended)

Ayers discloses

- An apparatus for compiling a computer program, the apparatus comprising:

  A processor configured to execute computer-readable code; a memory system configured to store data; computer-readable code for a front-end portion of compiler program, the front-end portion of the compiler program being configured to receive a plurality of

modules of source code (Fig.1), generate intermediate representations corresponding to the modules (Sec.2.1 1st Para. Lines 1-6), and extract a set of data from the intermediate representations to generate inliner summaries for the modules (sec.2.2 1st Para. Lines 1-4 & 3rd Para. Lines 1-5).

- after a call site is determined to be inlined: updating a call graph of the routines and call sites, and updating the inliner summaries throughout the call graph (Sec.2.3 The Last Para.).

Schmidt discloses

- Computer-readable code for a cross-module optimizer of the compiler program, the cross-module optimizer being configured to use the inliner summaries and a dynamically-updated globally-sorted working-list based order to analyze the call sites in an inline analysis phase, without using the intermediate representation, so as to determine which call sites in the modules are to be inlined by substituting code from a called module (col.3 lines 42~44 & lines 53-62).
- Determining the call sites to be inlined involves proceeding only once through the call sites (col.3 lines 42-53 *"A more global view of inlining effects is provided in order to select good inlining candidates while accurately spending the code bloat budget. First the best call sites at which to inline are estimated, based upon the execution frequencies of the call sites and the sizes of the called procedures. Then the call graph is processed starting from the leaves and working up. Each time an arc that was selected for inlining is encountered, the original bloat estimate is compared with the current size of the procedure, for example, incorporating sizes of any procedures that were inlined into the*

procedure. If the called procedure has been bloated beyond an acceptable limitation, it may be rejected for inlining.").

- dynamically updated the working list of call sites during the incline analysis phase (col.7 lines 31-38 *"If the call site exceeds the allowable growth at decision block 408, its priority is recalculated based on the bloat now known that will be incurred as indicated at a block 410. A best call site j residing in the AuxQueue is obtained as indicated at a block 412. Then checking to see whether the best call site j residing in the AuxQueue obtained at block 412 has a priority at least as great as the one being considering is performed as indicated at a decision block 414. "*).

11. Per claim 14 (Previously presented)

the rejection of claim 8 is incorporated and Ayers further discloses

- the inline analysis phase is separate and distinct from an inline transformation phase (Sec. 2.3 1st & 2nd Para. (i.e. analysis phase for determining which call site is clonable); Sec.2.3 6th & 7th Para. (i.e. transformation phase for using the results of analysis phase to create clones and fix call sites).

12. Per claim 15 (Currently amended)

Ayers discloses

- after a call site is determined to be inlined: updating a call graph of the routines and call sites, and updating the inliner summaries throughout the call graph (Sec.2.3 The Last Para.).

Schmidt discloses

- a computer program product comprising a computer-usable medium having computer-readable code embodied therein, the computer program product being compiled from a plurality of modules of source code using inliner summaries and a dynamically updated globally-sorted working-list based order in an inline analysis phase, without using intermediate representations, to determine which call sites in the modules are to be inlined by substituting code from a called module (col.3 lines 42~44 & lines 53-62).
- Determining the call sites to be inlined involves proceeding only once through the call sites (col.3 lines 42-53 *"A more global view of inlining effects is provided in order to select good inlining candidates while accurately spending the code bloat budget. First the best call sites at which to inline are estimated, based upon the execution frequencies of the call sites and the sizes of the called procedures. Then the call graph is processed starting from the leaves and working up. Each time an arc that was selected for inlining is encountered, the original bloat estimate is compared with the current size of the procedure, for example, incorporating sizes of any procedures that were inlined into the procedure. If the called procedure has been bloated beyond an acceptable limitation, it may be rejected for inlining."*).
- dynamically updated the working list of call sites during the incline analysis phase (col.7 lines 31-38 *"If the call site exceeds the allowable growth at decision block 408, its priority is recalculated based on the bloat now known that will be incurred as indicated at a block 410. A best call site j residing in the AuxQueue is obtained as indicated at a block 412. Then checking to see whether the best call site j residing in the AuxQueue*

obtained at block 412 has a priority at least as great as the one being considering is performed as indicated at a decision block 414. ").

13. Per claim 16 (Currently amended)

Schmidt discloses

A method of compiling a computer program with a plurality of modules of source code and corresponding intermediate representations (see Fig.1), the method comprising:

- using the inliner summaries in a one-pass inline analysis phase, without using the intermediate representations, to determine which call sites to inline, in what order to inline them, and preserving a same order during the transformation phase (col.2 lines 30~36 "*A first approximation of initial call sites of the identified possible call sites are identified for inlining. Procedures in the call multigraph are processed in a determined order where a first procedure is only processed after all second procedures called by the first procedure are processed.*").
- formulating a measure of goodness for each call site from the inliner summary (col.3 lines 40-44).
- using a technique to dynamically update information in the summary for potentially all call-graph nodes and edges every time a call site is accepted for inlining (col.3 lines 45-50).
- using a globally sorted work-list and an associated table to maintain and manipulate the call sites and dynamically updating the work-list every time a call site is accepted for inlining (col.3 lines 52-62).

Ayers discloses

- extracting a set of data from the intermediate representations of the modules to create an inliner summary for each module (sec.2.2 1st Para. Lines 1-4 & 3rd Para. Lines 1-5).

14. Per claim 17 (Currently amended)

the rejection of claim 16 is incorporated and Schmidt further discloses

- wherein the inliner summaries are comprised of: a code size (col.6 lines 3~4); a call site profile count (col.5 lines 30-33); a critical path length; an execution time (col.3 lines 44-46); a node level; a level criticality; and a total execution count (col.4 lines 30~32).

15. Per claim 18

- the rejection of claim 16 is incorporated and Schmidt further discloses
  wherein an arbitrary inlining order among the call sites in the call graph is selectable in an inline analysis phase, and wherein that same order is followed in an inline transformation phase (col.4 lines 2-6 "*the fundamental concept of the inline candidate selection method is to select an initial set of inline candidates and adaptively modify the initial set of inline candidates as more information becomes available.*").

16. Per claim 19

- the rejection of claim 16 is incorporated and Schmidt further discloses
  wherein a measure of goodness for each call site is computed from the light-weight inliner summary, and a total profit is computed as a product of component profit factors,

and a total cost is computed as a product of component cost factors (col.5 lines 21-25 "*The amount of code bloat that appropriately can incur is calculated by adding the total estimated instruction stream sizes for all procedures and multiplying by a tunable percentage factor, bloat factor as indicated at a block 202.*").

17. Per claim 20

- the rejection of claim 16 is incorporated and Schmidt further discloses wherein information in the light-weight inliner summary corresponding to call-graph nodes and edges are updated each time a call site is accepted for inlining, and wherein a goodness factor of each call site with updated summary information is re-computed (col.7 lines 31-33).

18. Per claim 21

- the rejection of claim 16 is incorporated and Schmidt further discloses wherein a globally-sorted work list and an associated table are maintained and used to continuously order the work list and extract a call site with a highest goodness factor (col.6 lines 8-13)

19. Claims 3, 6, 10, and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over by Ayers and in view of Schmidt (US Patent No. 6,195,793 B1).

20. Per claims 3 and 10 (Previously presented)

the rejection of claim 1 and 8 are incorporated respectively

Ayers does not teach

- after the call graph and inliner summaries are updated, re-calculating profitabilities associated with remaining call sites; and re-ordering the working list using the re-calculated profitabilities.

But Schmidt teaches

- re-calculating profitabilities associated with remaining call sites; and re-ordering the working list using the re-calculated profitabilities (Schmidt, col.7 lines 31-65).
- Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ayer's teachings by adding re-calculating profitabilities associated with remaining call sites; and re-ordering the working list using the re-calculated profitabilities as taught by Schmidt in order to determine whether the alternate call site from working list (i.e. AuxQueue) should be inlined if the priority best call site in AuxQueue is less than a threshold that is acceptable for the original call site after the re-calculating (Schmidt, col.7 lines 39-51).

21. Per claims 6 and 13 (Previously presented)

the rejection of claim 4 and 11 are incorporated respectively

Ayers does not teach

- the node summaries include at least a code size, a routine execution count, and a call-graph height.

But Schmidt teaches

- the node summaries include at least a code size, a routine execution count, and a call-graph height (Schmidt, col.4 lines 26-34).
- Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ayer's teachings by adding the node summaries include at least a code size, a routine execution count, and a call-graph height as taught by Schmidt in order to select good inlining and making accurate estimates of code bloat (Schmidt, col.3 lines 42-50).

Response to Arguments

Applicant's arguments filed on Mar. 14, 2008 have been fully considered but they are not persuasive.

In the remarks, Applicant argues that:

(a) In regard to claims 1, 8, and 15 the cited references do not teach "proceeding only once through the working list of call sites".

(b) In regard to claims 1, 8, and 15 the cited references do not teach "dynamically updating the working list of call sites".

Examiner responses:

Examiner disagrees.

(a) Schmidt implicitly discloses "determining the call sites to be inlined involves proceeding only once through the call sites" (col.3 lines 42-53 *"A more global view of inlining effects is provided in order to select good inlining candidates while accurately spending the code bloat budget. First the best call sites at which to inline are estimated, based upon the execution*

frequencies of the call sites and the sizes of the called procedures. Then the call graph is processed starting from the leaves and working up. Each time an arc that was selected for inlining is encountered, the original bloat estimate is compared with the current size of the procedure, for example, incorporating sizes of any procedures that were inlined into the procedure. If the called procedure has been bloated beyond an acceptable limitation, it may be rejected for inlining."). For each call site which determines to be inlined, it is processed through the procedures in the foregoing descriptions. This claim limitation which is supported in specifications on pages 19-20 does not specific disclosed "proceeding only once". Thus, examiner interpreted this limitation as cited paragraphs of Schmidt above.

(b) Schmidt implicitly discloses "dynamically updated the working list of call sites during the inline analysis phase" (col.7 lines 31-38 *"If the call site exceeds the allowable growth at decision block 408, its priority is recalculated based on the bloat now known that will be incurred as indicated at a block 410. A best call site j residing in the AuxQueue is obtained as indicated at a block 412. Then checking to see whether the best call site j residing in the AuxQueue obtained at block 412 has a priority at least as great as the one being considering is performed as indicated at a decision block 414. "*). If the call site exceeds a condition, its priority list will be recalculated. i.e. the priority will be updated depending upon the certain condition during inline analysis phase.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Junchun Wu whose telephone number is 571-270-1250. The examiner can normally be reached on 8:00-17:00 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JW

/Wei Zhen/

Supervisory Patent Examiner, Art Unit 2191